REMARKS

Claims 1-29 are pending in this application. Claims 1, 6, 8, 13, and 22-25 are independent claims. Claims 1, 6, 11, 13, 18, 22-27 and 29 are amended.

Claim Objections

Claims 1, 11, and 25 are objected due to minor informalities.

Applicants respectfully submit claims 1, 11 and 25 have each been amended in accordance with e Examiner's suggestion for overcoming these objections.

Claims 26, 27 and 29 are objected to under 37 CFR 1.75 as being substantial duplicates of claims 3, 4, and 12, respectively. Applicants respectfully submit claims 2, 27 and 29 have each been amended to address this objection.

Rejections Under 35 U.S.C. §112

Claim 24 is rejected under 35 U.S.C. §112, second paragraph, as being indefinite. This rejection is respectfully traversed.

The Examiner suggests amending claim 24 to replace the term "CodeML" with the phrase "first code" in order to address the rejection. Applicants respectfully submit claim 24 has been amended in accordance with the Examiner's suggestion. Accordingly, Applicants respectfully submit claim 24 is not indefinite and does meet the requirements of 35 U.S.C. §112, second paragraph.

Therefore, Applicants respectfully request the rejection of claim 24 under 35 U.S.C. § 112, second paragraph be withdrawn.

Rejections Under 35 U.S.C. §101

Claims 22-25 are rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. This rejection is respectfully traversed.

The Examiner asserts claims 22-25 are directed to non-statutory subject matter. Accordingly, claims 22-25 have each been amended to recite "A non-transitory computer readable medium encoded with computer executable instructions . . .". Consequently, Applicants respectfully submit claims 22-25 as amended are directed to statutory subject matter as defined by §101.

Therefore, Applicants respectfully request the rejection of claims 22-25 under 35 U.S.C. §101 be withdrawn.

Rejections Under 35 U.S.C. §102 – Waldin

Claims 1-4, 22, 26, and 27 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent 6,052,531 to Waldin et al. ("Waldin"). This rejection is respectfully traversed.

Claim 1 recites: "initially forming, from an original piece of software including only source text, a hybrid form of the original software, formed in such a way that at least one part of the source text is compiled into at least one of a byte and binary code and at least one further part of the source text is converted into a code formulated in a meta markup language for at least one variation point". With respect to this limitation the Examiner references column 4, lines 17-35 and column 6, line 48 – column 7, line 25 of Waldin. Applicants note Waldin is directed to a method of providing incremental updates to software (Waldin: Abstract). Waldin teaches incremental updates using binary patching which replaces binary bits of software of a first version with binary bits that differ in a newer version (Waldin: col. 1, line 48-65). The portions of Waldin referenced by the Examiner discuss a binary patch file builder 120 which produces

binary patch files 122 (which Waldin also refers to as DeltaPackages), and an entity called a DeltaCatalog which can specify a number of different DeltaPackages 122 corresponding to different versions of software being updated. However, for at least the reasons discussed below, Applicants respectfully submit Waldin fails to teach each of the elements of claim 1 as is required to support a rejection under §102.

First, it is not clear what in Waldin the Examiner is interpreting as corresponding to "the original piece of software including only source text" recited in claim 1. Specifically, none of the portions of Waldin referenced by the Examiner appear to discuss source text. The software application 110 which Waldin discusses patching is already compiled application which is updated by switching out bits as is indicated by column 1, lines 48-65 of Waldin. Further, the Examiner does not appear to interpret the binary patch files/Delta packages 122 being source text. Accordingly, Applicants respectfully submit, the Examiner has not identified any element corresponding to the original software including only source text recited in claim 1. For at least this reason, Applicants respectfully submit Waldin fails to teach each of the elements recited in claim 1 as is required to support a rejection under §102.

Further, assuming the Examiner is interpreting some element of Waldin as teaching original software including only source text, Applicants respectfully submit the Examiner has not identified how or where Waldin teaches forming, from the original piece of software including only source text, "a hybrid form of the original software, formed in such a way that at least one part of the source text is compiled into at least one of a byte and binary code and at least one further part of the source text is converted into a code formulated in a meta markup language for at least one variation point". Specifically, Applicants note Waldin teaches the DeltaCatalog being defined using XML. Accordingly, Applicants assume the Examiner is interpreting the DataCatalog as corresponding to the "code formulated in a meta markup language"

recited in claim 1. However, nothing in Waldin teaches the DeltaCatalog being **converted into XML** from source text, as is required of the "code formulated in a meta markup language" recited in claim 1. Accordingly, Applicants respectfully submit the Examiner has not identified anything in Waldin corresponding to the "code formulated in a meta markup language".

Further, even if, *for the sake of argument*, Waldin teaches the DataCatalog being a code converted into a meta language, nothing in Waldin teaches the DataCatalog being part of a **hybrid piece of software** which includes, in addition to code in a meta language converted from source text, one of a byte and binary code compiled from source text. Accordingly, Applicants respectfully submit the Examiner has not identified anything in Waldin corresponding to the "hybrid form of the original software, formed in such a way that at least one part of the source text is compiled into at least one of a byte and binary code and at least one further part of the source text is converted into a code formulated in a meta markup language for at least one variation point" recited in claim 1.

For at least these reasons, Applicants respectfully submit, Waldin fails to teach each of the elements of claim 1 or any claims depending from claim 1 as is required to support a rejection under §102.

Further, Applicants respectfully submit claim 22 includes limitations similar to those discussed above with reference to claim 1. Accordingly, for at least the reasons discussed above with respect to claim 1, Applicants respectfully submit Waldin fails to teach each of the elements of claim 22 or any claims depending from claim 22 as is required to support a rejection under §102.

Therefore, Applicants respectfully request the rejection of claims 1-4, 22, 26, and 27 rejected under 35 U.S.C. §102(b) be withdrawn.

Rejections Under 35 U.S.C. §102 – Germon

Claims 6, 7, and 23 are rejected under 35 U.S.C. §102(b) as being anticipated by "Using XML as an Intermediate Form for Compiler Development," by Germon ("Germon"). This rejection is respectfully traversed.

Claim 6 recites "making a first code formulated in a meta markup language with language extensions formulated in at least one meta markup language available as the source code" and "converting the source code, via a transformation in accordance with transformation rules, into a second code formulated in the meta markup language without the language extensions formulated in the meta markup language". With respect to teaching these limitations the Examiner references Germon which discusses techniques for translating formal grammars into XML and translating XML into other formal grammars (Germon: Abstract).

However, for at least the reasons discussed below, Applicants respectfully submit Germon fails to teach each of the elements of claim 6 as is required to support a rejection under §102.

First, Applicants respectfully submit nothing in Germon teaches a markup language with language extensions. Applicants note, the Examiner asserts place holders discussed on page 5 of Germon represent language extensions. However, Applicants can find no support for this assertion in Germon. Specifically, nothing in Germon teaches that the place holders are XML language extensions. Accordingly, Applicants respectfully submit, Germon does not teach "making a first code formulated in a meta markup language with language extensions formulated in at least one meta markup language available as the source code" as claim 6 recites.

Next, even if, *for the sake of argument*, the place holders of Germon are some sort of XML language extension, Germon does not teach converting the XML code with the language extensions into XML code without the language extensions as the

limitations of claim 6 require. Applicants note the Examiner's assertion that this feature is taught by the filling in of place holders discussed in the last paragraph on page 5 of Germon. However, Applicants respectfully disagree with the Examiner's interpretation of Germon. Specifically, the last paragraph on page 5 of Germon discusses issues regarding cross referencing within an **output of an XML to non-XML translation.** Further, Germon discusses a backpatching process which includes writing placeholders into **the output** and replacing the placeholders with real values. Accordingly, if anything, Germon discusses writing and replacing place holders in the **non-XML** language output as a result of the translation discussed on page 5 of Germon, not in XML code. Accordingly, Germon still fails to teach "converting the source code, via a transformation in accordance with transformation rules, **into a second code formulated in the meta markup language without the language extensions formulated in the meta markup language**" (emphasis added) as claim 6 recites.

For at least these reasons, Applicants respectfully submit, Germon fails to teach each of the elements of claim 6 or any claims depending from claim 6 as is required to support a rejection under §102.

Further, Applicants respectfully submit claim 23 includes limitations similar to those discussed above with reference to claim 6. Accordingly, for at least the reasons discussed above with respect to claim 6, Applicants respectfully submit Germon fails to teach each of the elements of claim 23 or any claims depending from claim 23 as is required to support a rejection under §102.

Therefore, Applicants respectfully request the rejection of claims 6, 7, and 23 rejected under 35 U.S.C. §102(b) be withdrawn.

Rejections Under 35 U.S.C. §103 – Waldin and Tan

Claim 5 is rejected under 35 U.S.C. §103(a) as being unpatentable over Waldin in view of U.S. Patent 7,536,686 to Tan et al. ("Tan"). This rejection is respectfully traversed.

The deficiencies of Waldin are discussed above with respect to claim 1 and are relevant here because claim 5 depends from, and thus incorporates the limitations of, claim 1. Tan fails to remedy these deficiencies. Accordingly, the combination of Waldin and Tan fails to teach each of the limitations of claim 5. Consequently, a prima facie case of obviousness has not been established with respect to claim 5 as is required to support a rejection under §103.

Therefore, Applicant respectfully request the rejection of claim 5 under §103(a) be withdrawn.

Rejections Under 35 U.S.C. §103 – Waldin and Germon

Claims 8-10, 12-17, 19-21, 24, 25, 28 and 29 are rejected under 35 U.S.C. §103(a) as being unpatentable over Waldin in view of Germon. This rejection is respectfully traversed.

Claims 8, 24 and dependents

Applicants respectfully submit the combination of Waldin and Germon fails to teach or render obvious each of the limitations of claim 8 for at least the following reasons.

First, claim 8 recites "converting a source code, formulated in a first programming language, into a first code formulated in a meta markup language". With respect to this limitation the Examiner again references column 4, lines 17-35 and column 6, line 48 – column 7, line 25 of Waldin. As is noted above, the portions

of Waldin referenced by the Examiner discuss a binary patch file builder 120 which produces binary patch files/ DeltaPackages 122, and the DeltaCatalog which can specify a number of different DeltaPackages 122 corresponding to different versions of software being updated. Applicants further note Waldin teaches the DeltaCatalog being defined using XML. Accordingly, Applicants assume the Examiner is interpreting the DeltaCatalog as corresponding to the "first code formulated in a meta markup language" recited in claim 8. However, nothing in Waldin teaches **converting** any type of source code into the DataCatalog. Accordingly, the DataCatalog of Waldin cannot correspond with the first code formulated in a meta markup language" recited in claim 8. Accordingly, the Examiner has not identified where or how Waldin or Germon, alone or in combination teach or otherwise render obvious "converting a source code, formulated in a first programming language, into a first code formulated in a meta markup language" as claim 8 recites.

Next, claim 8 recites "transforming the second code into a second source code formulated in at least one of the first programming language and a different programming language, the first and the second source code differing in terms of their functionality". With respect to this limitation, the Examiner references column 5, lines 26-65 of Waldin and appears to assert that Waldin teaches transforming an intermediate code into a second source code, where the so-called intermediate code corresponds to the second recited in claim 8. Column 5, lines 26-65 of Waldin discuss a process of transforming an application of state F to an application of state T. However, the limitations of claim 8 require that the second code be "formulated in the meta markup language". Based on the Examiner's apparent interpretation of the first code recited in claim 8, it appears the Examiner is identifying XML as the meta markup language. However, nothing in column 5, lines 26-65 of Waldin teaches translating XML code into source code. To the contrary, XML code does not even

appear to be mentioned in this portion of Waldin. Accordingly, the Examiner has not identified where or how Waldin or Germon, alone or in combination teach or otherwise render obvious "transforming the second code into a second source code formulated in at least one of the first programming language and a different programming language, the first and the second source code differing in terms of their functionality" as claim 8 recites.

For at least these reasons, Applicants respectfully submit the combination of Waldin and Germon fails to teach each of the limitations of claim 8. Accordingly, Applicants respectfully submit a *prima facie* case of obviousness has not been established with respect to claim 8 or any claims depending from claim 8 as is required to support a rejection under §103.

Further, Applicants respectfully submit claim 24 includes limitations similar to those discussed above with reference to claim 8. Accordingly, for at least the reasons discussed above with respect to claim 8, Applicants respectfully submit a *prima facie* case of obviousness has not been established with respect to claim 24 or any claims depending from claim 24 as is required to support a rejection under §103.

Claims 13, 25 and dependents

Applicants respectfully submit the combination of Waldin and Germon fails to teach or render obvious each of the limitations of claim 13 for at least the following reasons. First, claim 13 recites "adding an item of information formulated in the meta markup language and influencing the subsequent program execution, via a transformation, to the first code in at least one of a substituting and non-substituting way and wherein in this way, a second code also formulated in the meta markup language is formed, the transformation being performed in accordance with transformation rules formulated in a transformation description language".

Accordingly, the limitations of claim 1 require transforming first code, which is formulated in a meta markup language, into second code, which is also formulated in the meta mark up language, by adding an item of information to the first code. With respect to this limitation, the Examiner references column 4, lines 17-35 and column 6, line 48- column 7, line 25. As is noted above, the portions of Waldin referenced by the Examiner discuss a binary patch file builder 120 which produces binary patch files/ DeltaPackages 122 the DeltaCatalog which can specify a number of different DeltaPackages 122 corresponding to different versions of software being updated.

However, nothing in the portion of Waldin identified by the Examiner teaches transforming a first code formulated in a meta markup language into a second code formulated in the meta markup language by adding an item of information to the first code. Specifically, Applicants assume the Examiner is identifying the DeltaCatalog implemented in XML as corresponding to the first code recited in claim 13. However, Waldin does not teach performing any sort of transformation of the DeltaCatalog. Accordingly, Applicants respectfully submit the Examiner has not identified where Waldin or Germon, alone or in combination, teach or otherwise render obvious "adding an item of information formulated in the meta markup language and influencing the subsequent program execution, via a transformation, to the first code in at least one of a substituting and non-substituting way and wherein in this way, a second code also formulated in the meta markup language is formed, the transformation being performed in accordance with transformation rules formulated in a transformation description language" as claim 13 recites. For at least this reason, Applicants respectfully submit a *prima facie* case of obviousness has not been established with respect to claim 13 or any claims depending from claim 13 as is required to support a rejection under §103.

Next, for the following additional reason, Applicants respectfully submit the Examiner has not established a *prima facie* case of obviousness with respect to claim 13 as is required to support a rejection under §103.

The following has been held

> "rejections on obviousness cannot be sustained with mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006). See also *KSR*, 550 U.S. at ___ , 82 USPQ2d at 1396 (quoting Federal Circuit statement with approval)

Accordingly, Applicants respectfully submit the Examiner must articulate some reasoning having a rational underpinning supporting the conclusion of obviousness with respect to claim 13. Applicants respectfully submit the Examiner has not articulated such a reasoning.

Specifically, the Examiner's asserts that Germon teaches converting a source code formulated in a first programming language into a first code formulated in a meta mark up language, and that it would be obvious to combine the teachings of Germon with those of Waldin to teach the limitations of claim 13. With respect to the reasoning behind this conclusion, the Examiner asserts that Germon includes motivation for the combination. Specifically, with respect to motivation, the Examiner cites portions of Germon discussing why XML was used by Germon as an intermediate language for the compiler implementation to which Germon is directed. However, it is not clear how the portions of Germon referenced by the Examiner would motivate a person of ordinary skill in the art to modify Waldin. Specifically, Waldin is not directed to compiler implementation like Germon. Waldin is directed to software updating (Waldin: Abstract). Accordingly, the Examiner has identified **no benefit** for a person of ordinary skill in the art using the software updating system of Waldin that would be realized by making the combination proposed by the Examiner.

Further, even if, *for the sake of argument*, the portions of Germon cited by the Examiner can be interpreted as disclosing benefits of the teachings of Germon, the Examiner does not explain how benefits relating to compiler implantation, discussed in the portion of Germon identified by the Examiner, are in any way relevant to the software updating system to which Waldin is directed. Accordingly, Applicants respectfully submit the Examiner has not identified anything in Waldin, Germon or any other source that would actually **motivate** one of ordinary skill in the art to modify the software updating system of Waldin to somehow include the teachings of Germon. Accordingly, Applicants respectfully submit the Examiner has not identified a reasoning having a rational underpinning supporting the conclusion of obviousness with respect to claim 13 as is required to support a rejection under §103.

Further, Applicants respectfully submit claim 25 includes limitations similar to those discussed above with reference to claim 13. Accordingly, for at least the reasons discussed above with respect to claim 13, Applicants respectfully submit a prima facie case of obviousness has not been established with respect to claim 25 or any claims depending from claim 25 as is required to support a rejection under §103.

Rejections Under 35 U.S.C. §103 – Waldin, Germon and Tan

Claims 11 and 18 are rejected under 35 U.S.C. §103(a) as being unpatentable over Waldin in view of Germon and Tan. This rejection is respectfully traversed.

The deficiencies of Waldin and Germon are discussed above with respect to claims 8 and 13 and are relevant here because claims 11 and 18 depend from, and thus incorporate the limitations of, claims 8 and 13, respectively. Tan fails to remedy these deficiencies. Accordingly, the combination of Waldin, Germon and Tan fails to teach or render obvious each of the limitations of either of claims 8 and 13.

Consequently, a *prima facie* case of obviousness has not been established with respect to either of claims 8 and 13 as is required to support a rejection under §103.

Therefore, Applicant respectfully request the rejection of claim 5 under §103(a) be withdrawn.

CONCLUSION

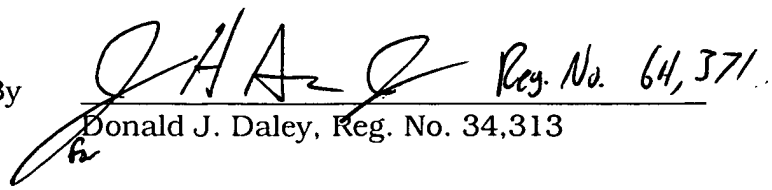
Accordingly, in view of the above amendments and remarks, reconsideration of the objections and rejections and allowance of each of the pending claims in connection with the present application is earnestly solicited.

Should there be any outstanding matters that need to be resolved in the present application, the Examiner is respectfully requested to contact the undersigned at the telephone number below.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 08-0750 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. §1.17; particularly, extension of time fees.

Respectfully submitted,

HARNESS, DICKEY, & PIERCE, P.L.C.

By  Reg. No. 64,371

Donald J. Daley, Reg. No. 34,313

P.O. Box 8910
Reston, Virginia 20195
(703) 668-8000

DJD/JHA:eaf

1090923.1